ONNX Runtime-ZenDNN Windows User Guide

Revision #	**4.0**
Issue Date	**January 2023**

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

Trademarks

Dolby Laboratories, Inc.

Manufactured under license from Dolby Laboratories.

Rovi Corporation

This device is protected by U.S. patents and other intellectual property rights. The use of Rovi Corporation's copy protection technology in the device must be authorized by Rovi Corporation and is intended for home and other limited pay-per-view uses only, unless otherwise authorized in writing by Rovi Corporation.

Reverse engineering or disassembly is prohibited.

USE OF THIS PRODUCT IN ANY MANNER THAT COMPLIES WITH THE MPEG-2 STANDARD IS EXPRESSLY PROHIBITED WITHOUT A LICENSE UNDER APPLICABLE PATENTS IN THE MPEG-2 PATENT PORTFOLIO, WHICH LICENSE IS AVAILABLE FROM MPEG LA, L.L.C., 6312 S. FIDDLERS GREEN CIRCLE, SUITE 400E, GREENWOOD VILLAGE, COLORADO 80111.

Contents

List of Figures

List of Tables

Revision History

Date	Revision	Description
January 2023	4.0	Initial version.

Chapter 1 Installing ZenDNN with ONNX Runtime

Note: *You must refer ZenDNN v4.0 User Guide before starting the installation.*

1.1 Binary Release Setup

1.1.1 Conda

Complete the following steps to setup Conda:

1. Refer to Anaconda documentation (*https://docs.anaconda.com/anaconda/install/windows/*) to install Anaconda on your system.
2. Create and activate a Conda environment which will house all the ONNX Runtime-ZenDNN specific installations:

   ```
   conda create -n onnxrt-v1.12.1-zendnn-v4.0-rel-env python=3.8

   conda activate onnxrt-v1.12.1-zendnn-v4.0-rel-env
   ```

   Ensure that you install the ONNX Runtime-ZenDNN package corresponding to the Python version with which you created the Conda environment.
3. It is recommended to use the naming convention:

   ```
   onnxrt-v1.12.1-zendnn-v4.0-rel-env
   ```

4. Install all the necessary dependencies:

   ```
   pip install -U cmake numpy==1.23.2 pytest psutil torch==1.10.0 coloredlogs

   pip install -U transformers sympy --ignore-installed ruamel.yaml

   pip install onnx==1.12.0

   pip install protobuf==3.20.1
   ```

   Note: *For binary packages built with Python v3.7, it is recommended to use numpy v1.21.6 (numpy==1.21.6).*
5. Download AOCL-BLIS from AMD Developer Central (*https://developer.amd.com/amd-aocl/*).
6. Add BLIS path to the environment variable “Path”. For example, *C:\amd-blis\lib\ILP64*.
7. Download and install LLVM (Windows 64-bit) for *libomp.dll* (OpenMP: used for parallel programming) from GitHub (*https://github.com/llvm/llvm-project/releases/tag/llvmorg-14.0.6*).
8. Add *libomp.dll*, *libiomp5d.dll* path to the environment variable “Path”. For example, *C:\Program Files\LLVM\lib*.

1.1.2 ONNX Runtime v1.12.1

Complete the following steps to install the ZenDNN binary release:

1. Copy the zipped release package to the local system being used. The name of the release package will be similar to *ONNXRT_v1.12.1_ZenDNN_v4.0_Python_v3.8_Win.zip*.
2. Execute the following commands:
   a. `unzip ONNXRT_v1.12.1_ZenDNN_v4.0_Python_v3.8_Win.zip`
   b. `cd onnxrt-v1.12.1-ZenDNN-4.0-Python_v*/`
   c. `call scripts/zendnn_ONNXRT_env_setup_win.bat`
      This script will set up the required environment to run ONNX Runtime in optimal mode.
   d. `python -m pip install <whlfile.whl>`
   e. `pip install protobuf==3.20.1`

   Notes:

   1. Ensure that it is sourced only from the unzipped release folder.

   2. If there is any conda environment named onnxrt-1.12.1-zendnn-v4.0-rel-env already present, delete the conda environment onnxrt-1.12.1-zendnn-v4.0-rel-env (using command conda remove --name onnxrt-1.12.1-zendnn-v4.0-rel-env --all) before running scripts/zendnn_ONNXRT_env_setup_win.bat.

Chapter 2 Directory Structure

The release folder consists of a ONNXRT wheel (.whl), LICENSE and THIRD-PARTY-PROGRAMS files, and the following directories:

- *scripts* contains scripts to set up the environment

Chapter 3 High-level Overview

The following is a high-level block diagram for the ZenDNN library, which uses the AOCL-BLIS library internally:

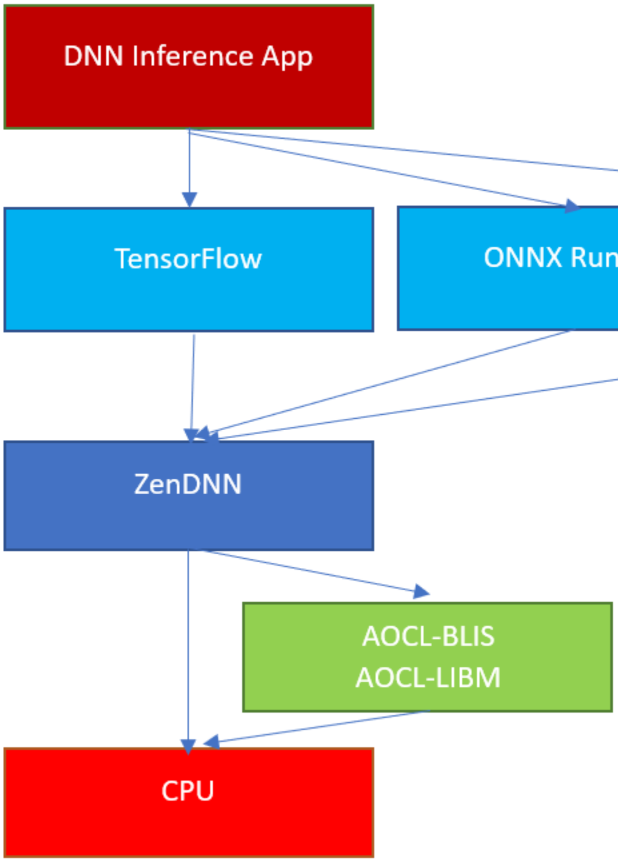

Figure 1. ZenDNN Library

In the current release, ZenDNN is integrated with TensorFlow, PyTorch, and ONNX Runtime.

Chapter 4 Environment Variables

ZenDNN uses the following environment variables to setup paths and control logs:

Table 1. ZenDNN Environment Variables-Generic

Environment Variable	Default Value/User Defined Value
ZENDNN_LOG_OPTS	ALL: 0
ZENDNN_PARENT_FOLDER	Path to unzipped release folder
ZENDNN_PRIMITIVE_CACHE_CAPACITY	The default value is set to 1024, you can modify it as required[a]
OMP_DYNAMIC	FALSE

[a] These environment variables work only for Blocked Format.

The following is a list of environment variables to tune performance:

Table 2. ZenDNN Environment Variables-Optimization

Environment Variable	Default Value/User Defined Value
OMP_NUM_THREADS	The default value is set to 64. You can set it as per the number of cores in the user system[a].
OMP_WAIT_POLICY	ACTIVE
OMP_PROC_BIND	FALSE
ZENDNN_CONV_ADD_FUSION_ENABLE	The flag is to enable convolution and add operator fusion. It is disabled (set to 0) by default. You can modify it to 1 to enable the fusion.
ZENDNN_RESNET_STRIDES_OPT1_ENABLE	The flag is to enable strides trick optimization for Resnet blocks. It is disabled (set to 0) by default. You can modify it to 1 to enable the optimization.
ZENDNN_BN_RELU_FUSION_ENABLE ZENDNN_CONV_CLIP_FUSION_ENABLE ZENDNN_CONV_RELU_FUSION_ENABLE ZENDNN_CONV_ELU_FUSION_ENABLE ZENDNN_GEMM_ALGO=3	This flag is disabled by default. You can use set command in Windows to set it to 1 and enable it.

Note: *There are a few other environment variables that are initialized by the setup script, however these are not applicable for the binary release setup.*

When source *scripts/zendnn_ONNXRT_env_setup_win.bat* is invoked, the script initializes all the environment variables except the one(s) which must be set manually. The environment variable **ZENDNN_PARENT_FOLDER** is initialized relative to the path defined by the unzipped release folder. To ensure that the paths are initialized correctly, it is important that the script is invoked from the unzipped release folder.

Chapter 5 Tuning Guidelines

The hardware configuration, OS, Kernel, and BIOS settings play an important role in performance. The details for the environment variables used on a 4th Gen AMD EPYC™ server to get the best performance numbers are as follows:

5.1 System

A system with the following specifications has been used:

Table 3. System Specification

Processor	AMD Ryzen™ Threadripper™ PRO 3995WX
RAM	512 GB
Socket	1
Physical Core	64
SMT: Thread(s) per Core	2
ONNXRT Version	1.12.1
ZenDNN Version	4.0

5.2 Environment Variables

The following environment variables have been used:

ZENDNN_LOG_OPTS=ALL:0

OMP_NUM_THREADS=64

OMP_WAIT_POLICY=ACTIVE

OMP_PROC_BIND=FALSE

OMP_DYNAMIC=FALSE

ZENDNN_GEMM_ALGO=3

ZENDNN_PARENT_FOLDER=/home/<user_id>/my_work

BENCHMARKS_GIT_ROOT=/home/<user_id>/my_work/benchmarks

ZENDNN_PRIMITIVE_CACHE_CAPACITY=1024

ZENDNN_ONNXRT_VERSION=1.12.1

ZENDNN_ONNX_VERSION=1.12.0

ZENDNN_CONV_ADD_FUSION_ENABLE=0

ZENDNN_RESNET_STRIDES_OPT1_ENABLE=0

As mentioned in “Environment Variables” on page 11, the script *scripts/zendnn_ONNXRT_env_setup_win.bat*, initializes all the environment variables except the one(s) which you must set manually. The environment variables **OMP_NUM_THREADS**, **OMP_WAIT_POLICY** and **OMP_PROC_BIND** can be used to tune performance. For optimal performance, the **Batch Size** must be a multiple of the total number of cores (used by the threads). On a 3rd Gen AMD Ryzen^{TM} Threadripper^{TM} workstation (configuration: AMD Ryzen^{TM} Threadripper^{TM} PRO 3995WX, 1P and **SMT=ON**) with the above environment variable values, **OMP_NUM_THREADS=64** yield the best throughput numbers for a single socket.

KMP_DUPLICATE_LIB_OK=TRUE is used to load multiple libomp instances.

Batch Size is a sensitive factor for the throughput performance of any model. The following formula could be used to calculate the optimal **Batch Size**:

Batch Size = number_of_physical_cores * batch_factor

batch_factor may vary from 8-32. Usually, the value 32 gives the optimal performance.

5.3 Optimal Setting

Optimal performance of several ZenDNN workloads is observed when interleaving is enabled in conjunction with the NPS4 mode.

By default, ONNX Runtime uses Visual Studio OpenMP (libomp) for parallel computation. For ZenDNN backend, you can download LLVM OpenMP’s libomp for better performance.

You can download:

- LLVM from GitHub (*https://github.com/llvm/llvm-project/releases/tag/llvmorg-14.0.6*)
- Visual Studio from Microsoft website (*https://learn.microsoft.com/en-us/visualstudio/releases/2019/history*).

Chapter 6 License

ZenDNN is licensed under Apache License Version 2.0. Refer to the "LICENSE" file for the full license text and copyright notice.

This distribution includes third party software governed by separate license terms.

3-clause BSD license:

- Xbyak (*https://github.com/herumi/xbyak*)
- Googletest (*https://github.com/google/googletest*)
- Instrumentation and Tracing Technology API (*https://github.com/intel/ittapi*)

Apache License Version 2.0:

- oneDNN (*https://github.com/oneapi-src/oneDNN*)
- Xbyak_aarch64 (*https://github.com/fujitsu/xbyak_aarch64*)

Boost Software License, Version 1.0:

Boost C++ Libraries (*https://www.boost.org/*)

MIT License from ONNXRT:

https://github.com/microsoft/onnxruntime

This third-party software, even if included with the distribution of the Advanced Micro Devices software, may be governed by separate license terms, including without limitation, third-party license terms, and open-source software license terms. These separate license terms govern use of the third-party programs as set forth in the THIRD-PARTY-PROGRAMS file.

Chapter 7 Technical Support

Please email *zendnnsupport@amd.com* for questions, issues, and feedback on ZenDNN.